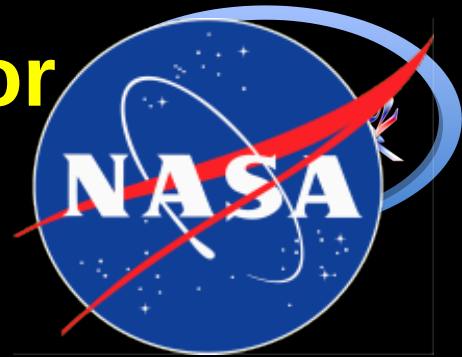


SEE Test and Data Analysis for Complex FPGA Systems



Methods for Mission-Critical Survivability Characterization

Melanie.D.Berg@SpaceR2.com

Acronyms



- Application specific integrated circuit (ASIC)
- Block random access memory (BRAM)
- Combinatorial logic (CL)
- Configurable Logic Block (CLB)
- Device under test (DUT)
- Edge-triggered flip-flops (DFFs)
- Field programmable gate array (FPGA)
- High speed serial interface (GTX)
- Input - output (I/O)
- Intellectual property (IP)
- INV (inverter)
- Linear energy transfer (LET)
- Look up table (LUT)
- Mean fluence to failure (MFTF)
- One time programmable (OTP)
- Operational frequency (f_s)
- Power on reset (POR)
- Place and Route (PR)
- Representative Tactical Design (RTD)
- Reprogrammable (RP)
- Single event functional interrupt (SEFI)
- Single event effects (SEEs)
- Single event failure (SEF)
- Single event latch-up (SEL)
- Single event transient (SET)
- Single event upset (SEU)
- Single event upset cross-section (σ_{SEU})
- Static random access memory (SRAM)
- Static timing analysis (STA)
- System on a chip (SOC)

Survivability for Mission Critical Applications: Problem Statement



Single event upset (SEU)
Field programmable gate
array (FPGA)

For SEU analysis, common practice is to use simple test structures that focus on discrete components:

- Data are extrapolated into survivability calculators.
- Generic SEU data are used across all designs.
- Assumption: the need for testing is reduced.
- However, the fidelity of generic SEU data extrapolation to tactical designs is questionable.

Better to use representative tactical designs (RTD) for SEU analysis:

- Data are a better fit for characterizing tactical behavior.
- However, requires SEU testing for every design!



How do we provide SEU data for survivability calculations of tactical systems; while reducing the need to test every design? Generic testing versus Test-As-You-Fly.

Additional Notes on Generic SEE Data versus RTD/MFTF SEE Data



NEPP: NASA Electronics Packaging Program

RTD: Representative tactical design

- Most FPGA data provided to the community are component level/generic... methodology was derived from traditional means of characterization from the space community.
- Organizations such as NEPP and FPGA manufacturers will always perform a component level investigation on FPGAs:
 - First look
 - Flush out
 - General idea if mitigation will be required
 - Important information for the community
- As FPGA devices become more complex extrapolation from simple component structures to RTD is not an appropriate method for tactical characterization.
 - **This is especially true when a design contains user inserted mitigation.**
- Test-as-you-fly is not a new concept. NEPP has been performing test-as-you-fly (RTD/MFTF) FPGA SEE investigations for mission (program-specific experiments).

RTD Definition and Considerations



- RTD Definition: Test design that approximates the tactical (target) design.
- When we cannot test tactical, we test RTD.
- Considerations for using RTD versus tactical (SEE Testing):
 - Tactical is not available to test.
 - SEE designs require an increase in observation points.
 - Test fixtures can limit I/O, stimulus capability, and capture capability.
 - SEE experiments require the test system to force the DUT into desired states (increases test coverage and assists in validating dominant mechanisms of failure).



The test-as-you-fly methodology requires complex test systems with an abundance of observation and reporting mechanisms. Requires more than simple heartbeat monitoring.

Representative Tactical Design (RTD) Test Structures and MFTF Test Strategies



Mean fluence to failure (MFTF)

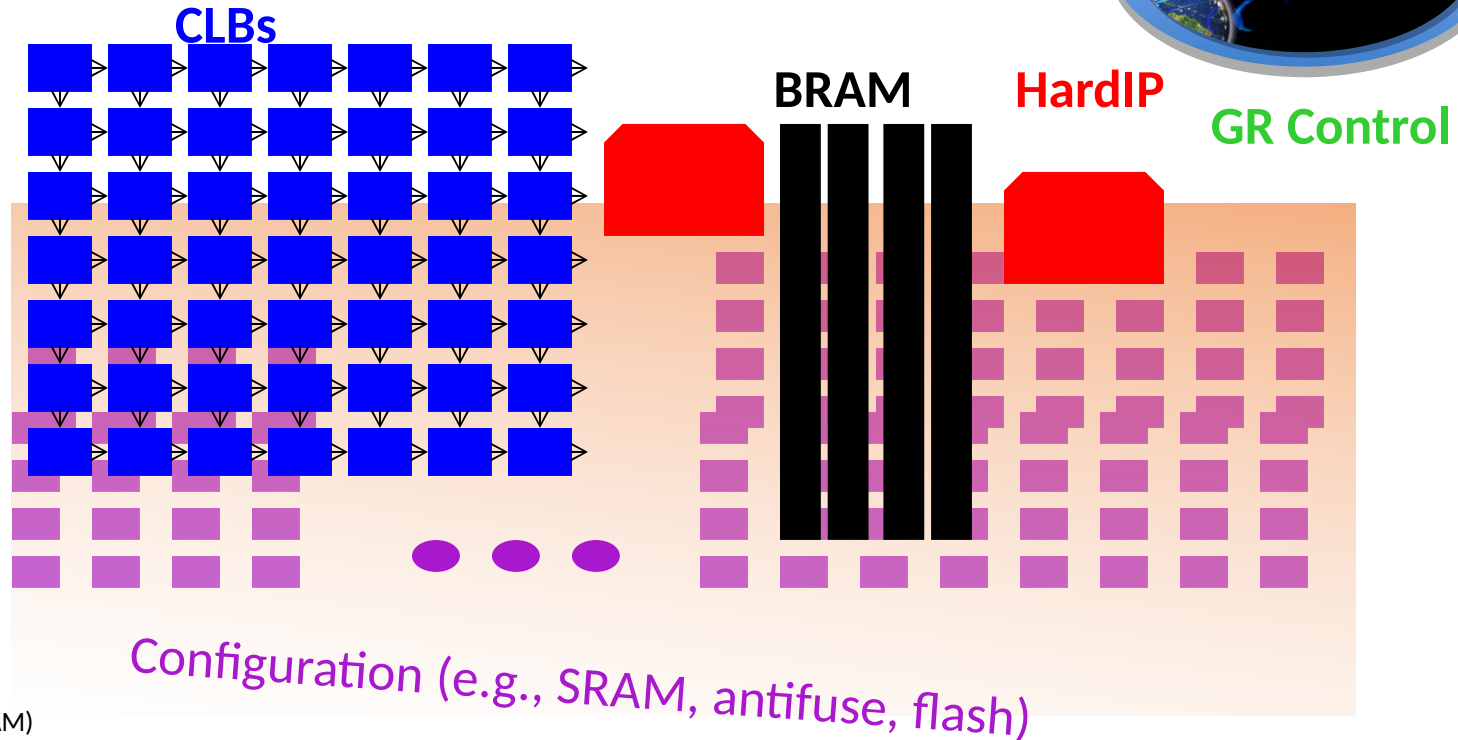
- **RTDs are based on tactical designs and might contain the following:**
 - Embedded processors
 - Highspeed serial (GTX)
 - Embedded SRAM (BRAM)
 - Global routes
- **Obey tactical design strategies:**
 - Synchronous design
 - Routing/floorplanning specifics
- **Piecemeal tests, yet use complex structures:**
 - Increases visibility
 - Study trends
 - Have at least one full RTD (close as possible to tactical)
- MFTF testing requires an increase in the number of experiments (statistics).
- MFTF testing will be driven by dominant mechanisms of failure in the design (given proper testing and visibility into failure).



FPGA SEU Cross Section Model



Complex routing
everywhere.



Configurable logic block: (CLB)

Block random access memory: (BRAM)

Intellectual property: (IP); e.g., micro processors, digital signal processor blocks (DSP), embedded state machines, etc.

Global Routes: (GR)

Analog circuits

SEU Cross sections for a mapped design () are based on the FPGA's internal elements and the mapped design's topology.

$$\sigma_{SEF} = f(\sigma_{configuration}, \sigma_{BRAM}, \sigma_{functionalLogic}, \sigma_{HiddenLogic})$$

Dominant mechanisms of failure will drive

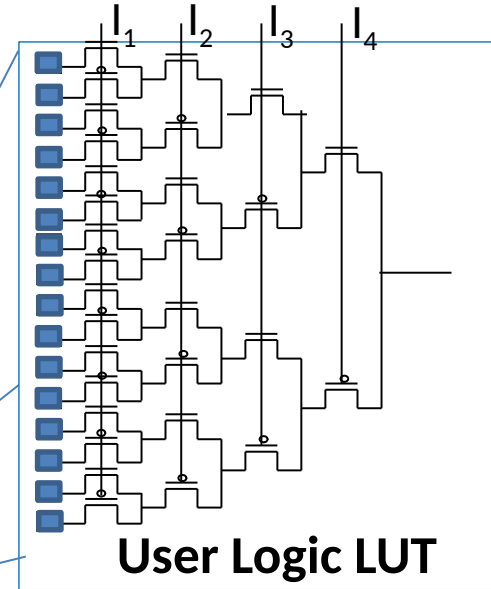
Embedded View of Mapped Logic



FPGA configuration and user logic are different types of embedded components.

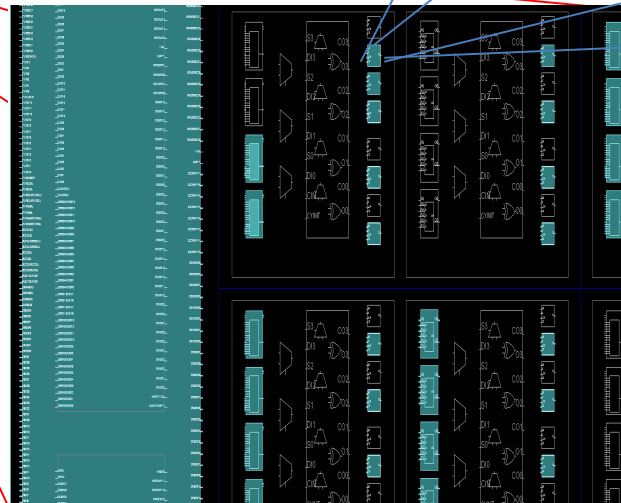


Configuration

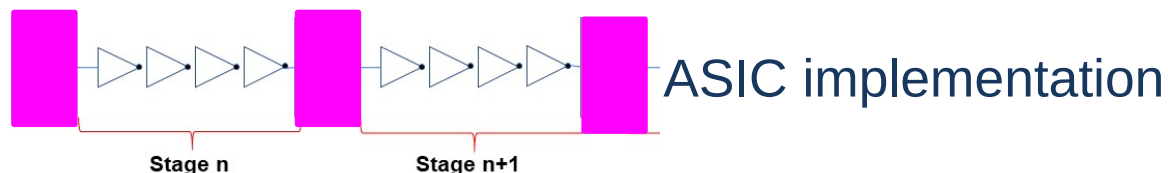


Designs only map into a portion of the configuration and only use a portion of the user fabric logic gates.

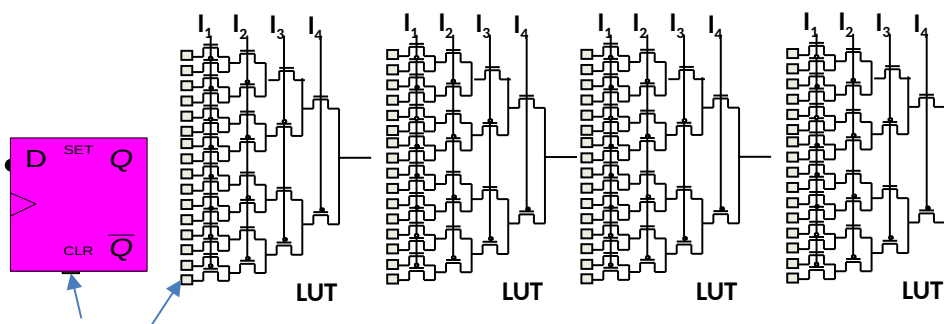
Modern FPGAs have 100's of millions of configuration bits and 100's of thousands of logic cells.



Generic Test Structures: Shift Register



User logic: Lookup Table (LUT)

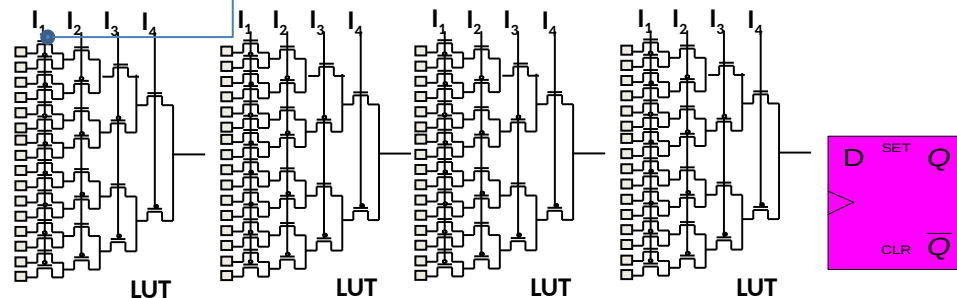


User logic: Flip-Flop(DFF)

LUTs and DFFs are contained in configuration logic blocks (CLBs)

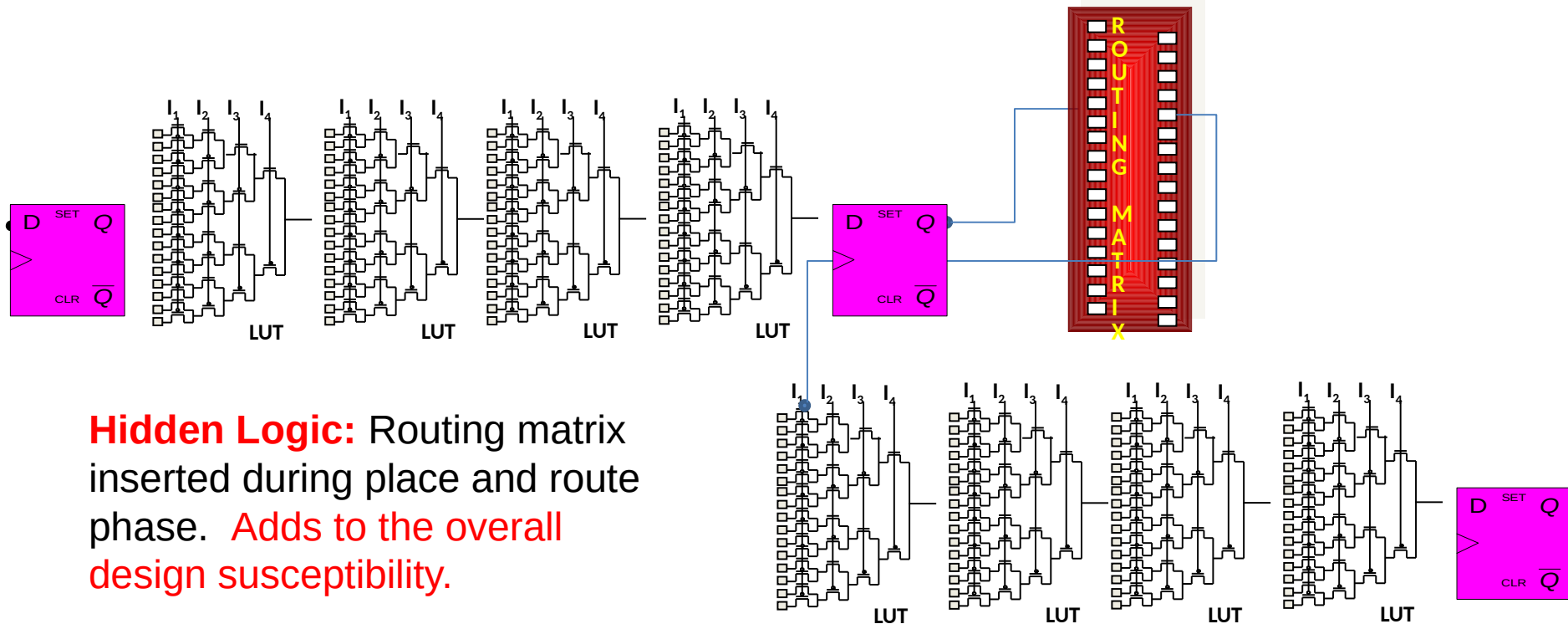
Configuration bits

With an SRAM-based FPGA, each design uses more logic than assumed. Makes extrapolation of SEU data (from simple test structures to tactical designs) unreliable.



Generic Xilinx Implementation
(LUT can differ by family)

Closer Look: Shift Register with Manufacturer Inserted Routing Matrix (Hidden Logic)



Simple test structures will not capture the impact of a tactical design's hidden logic (data are not extrapolatable). Hence the drive towards testing RTD structures.

RTD Test Structures and MFTF Strategies: Not a Simple Task



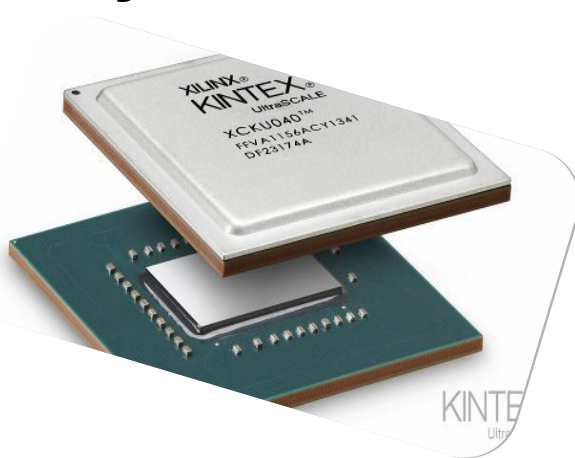
The following expertise is required:

- Professional design techniques
- Complex test system development
- The ability to create visibility into test structures for proper MFTF measurement
- Knowledge of test facilities





Data Analysis: Easing the process of SEU test and analysis for tactical-design survivability prediction.



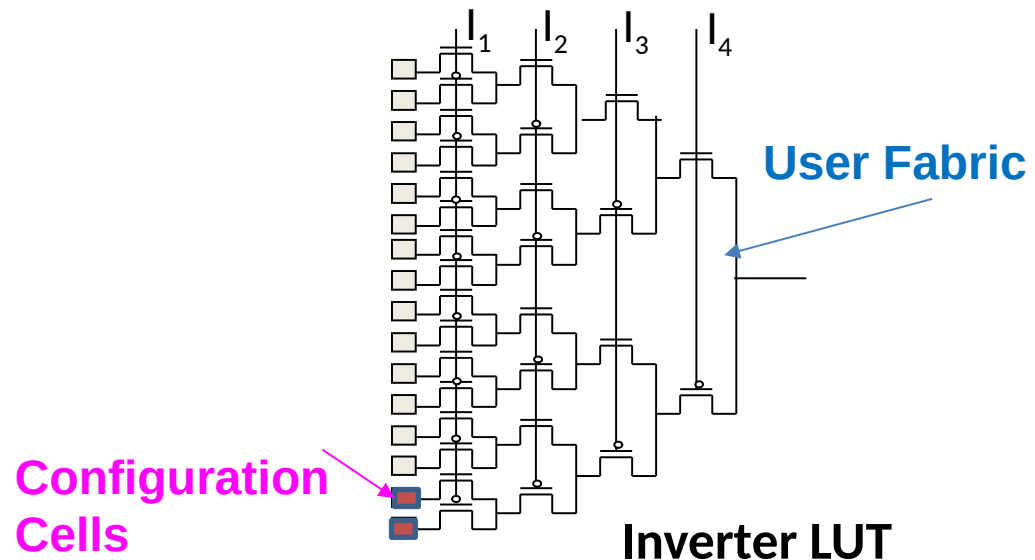
The following slides only apply to Xilinx SRAM-based FPGA devices with no embedded or user inserted mitigation.

Configuration, Mask, and Essential Bits



Design mapping into user fabric logic cells is defined by configuration bit settings.

- Configuration bits: Total number of configuration cells... (fixed per each FPGA type)
 - Masked bits: calculated by the manufacturer and is not under user control... design and device dependent
 - Unmasked bits
- Essential bits: number of configuration cells used by the design mapping (calculated by the manufacturer upon user directive... design and device dependent).





SEU Cross-Sections



- Cross-section Categorization:
 - Across all configuration cells (device)
 - Per configuration cell (device-bit)
 - Across essential-bits (Design + device)
 - Design specific

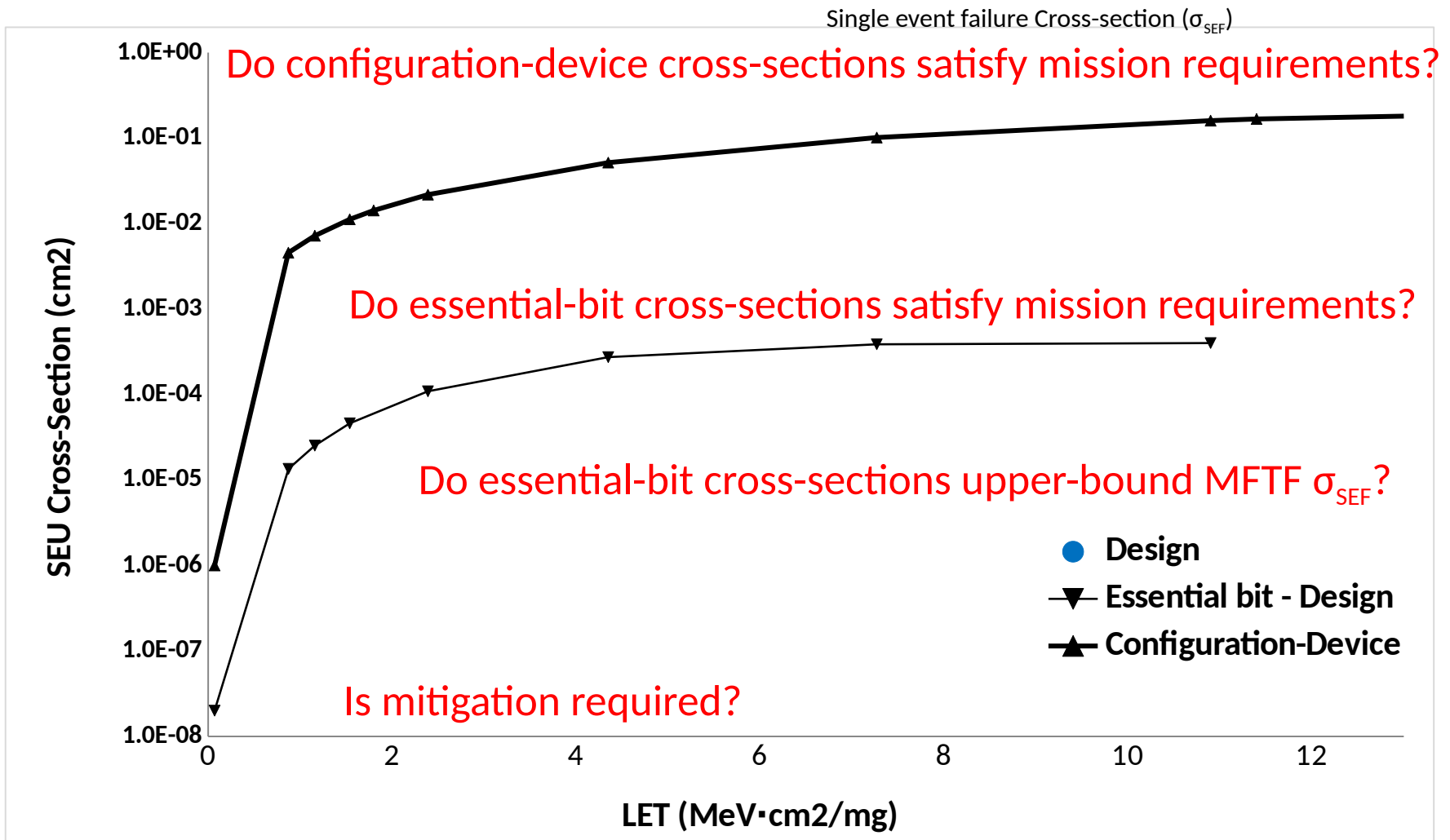
Generally, configuration cross-sections are readily available from generic device investigations.

$$\sigma(LET)_{Essential_{bit}} = Essential_{bits} \times \sigma(LET)_{configuration_{bit}}$$

*AverageFlux)

**Which cross-sections do we use for survivability analysis?
Must consider mission requirements.**

If Upper-bounds Satisfy Mission Reliability/Survivability Requirements, Then No Mitigation is Required.



Prove It Before You Use It



- Using as upper-bounds is not a new concept.
- However, should only be used if it is known/proven to be an upper-bound (or close enough depending on criticality).
- **The proof of bounding has been the missing factor; and is now necessary.**
- Why now? Device complexity includes a significant amount of hidden logic.
 - Hidden logic have components that are not included in the essential bit count.
 - It has shown (in flight) to impact susceptibility (e.g., internal scrubbers).



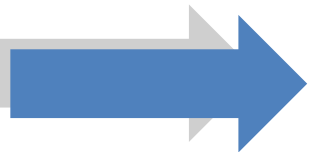
Xilinx SEU Test and Analysis: What Can the Manufacturer Provide?

Front-end Proof of Concept

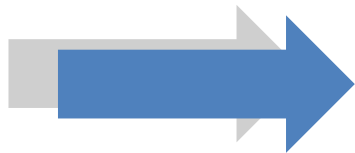


$$\sigma(LET)_{Essential_{bit}} = Essential_{bits} \times \sigma(LET)_{configuration_{bit}}$$

- Goal is to determine if generic data can be extrapolated to characterize complex tactical designs.
- Providing DFF, CLB, and LUT generic test data is not extrapolatable.
 - Topology effects are non-linear and does not include hidden logic.
- An alternative is to prove is an upper-bound to .



Manufacturer performs a variety of tests (benchmarks) to compare to .



Manufacturer provides generic data: configuration, BRAM, and embedded logic cross-sections.



Manufacturer performs additional testing to investigate potential SEFIs and other device SEE susceptibilities (global routes and SEL).

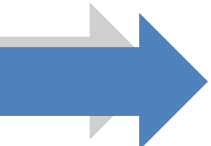
Xilinx SEU Test and Analysis: What Does The End-User Do with The Data?

Application of Concept




Intellectual property (IP)


- If proves to be a satisfactory upper-bound, the data and the tactical design's calculated essential-bits can be used by development teams for survivability analysis.
- In the past, has been assumed (by some) to be adequate for survivability prediction. However, as technology shrinks the need for RTD-MFTF testing and proof of concept is growing:
 - Mixed-signal circuitry, global-routes, and hidden logic (embedded IP cores) will have more impact on at low LETs.



Compare your design to manufacturer benchmark designs. Use for survivability calculations if >



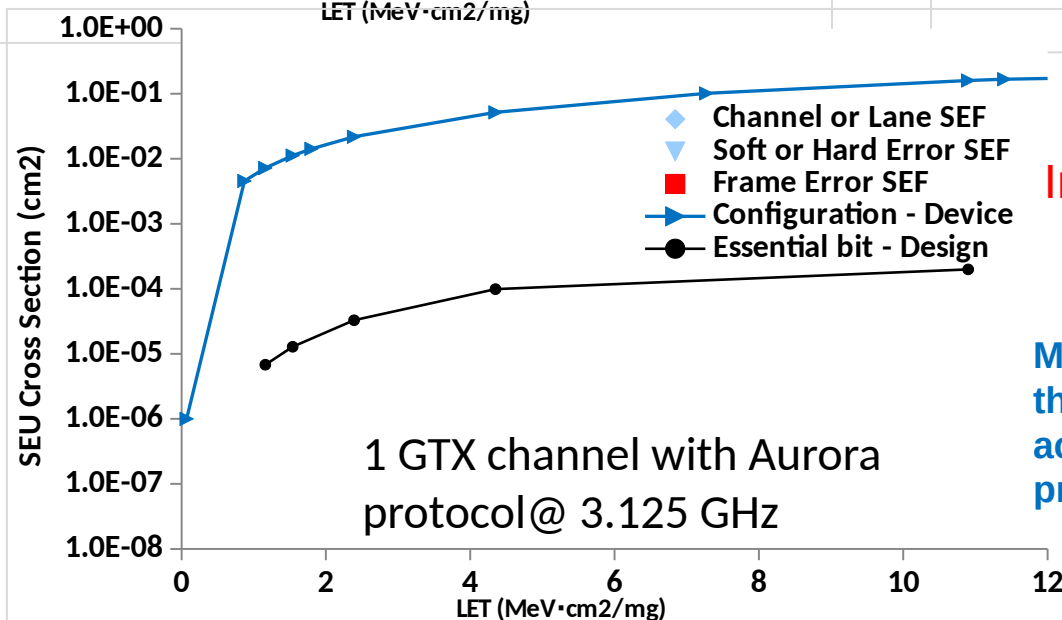
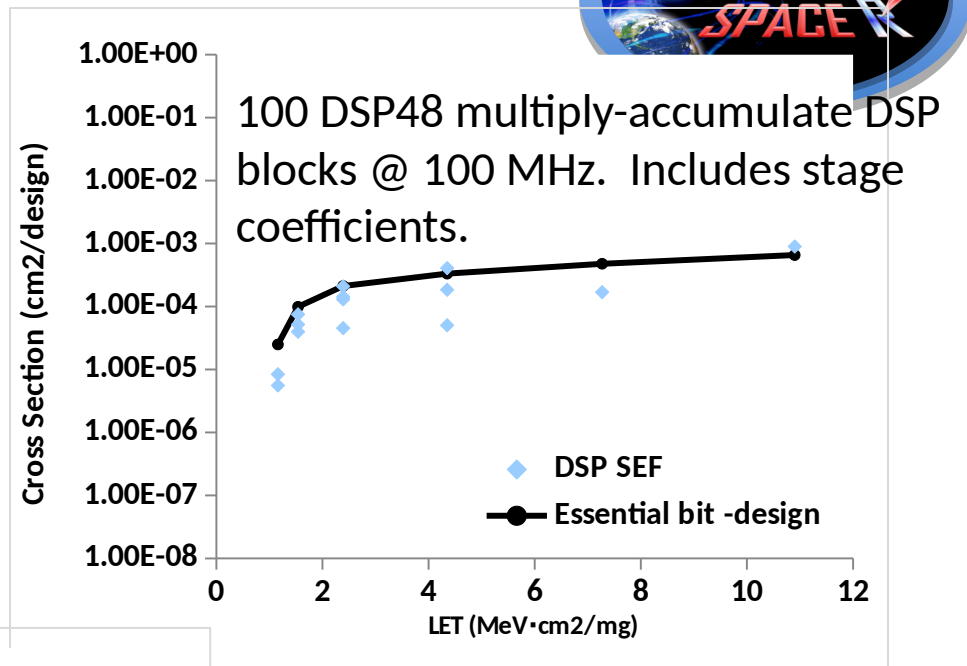
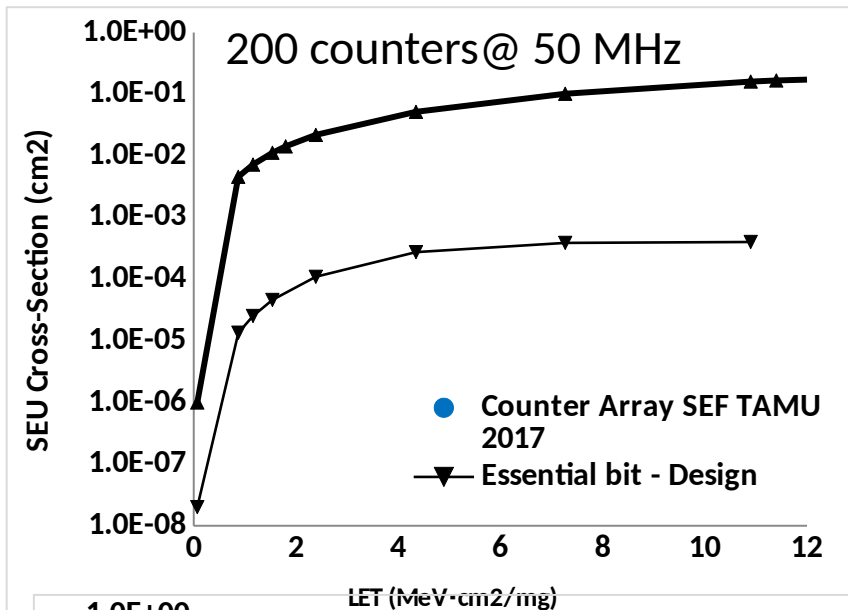
If manufacturer data show anomalies or your tactical design has untested complexities, additional RTD testing will be needed.



The end-user should not piecemeal small grained components (e.g., CLBs) for survivability analysis because of hidden logic and topological non-linearities.



Kintex UltraScale SEU Cross-Sections



$$\sigma_{\text{essential_bit}} > \sigma_{\text{SEF}}$$

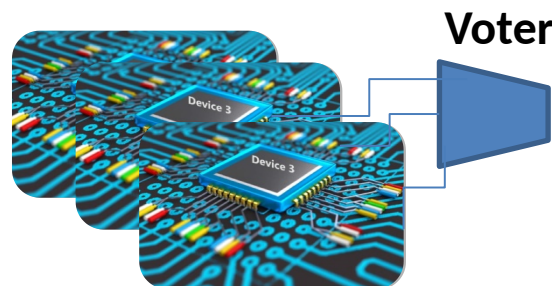
Implies $\sigma_{\text{Essential_bit}}$ can be used to predict survivability (non-mitigated design).

More testing will be performed to investigate if there are SEFIs and if upper-bound holds across complex designs (e.g., embedded processors); and higher LET.

Mitigation Analysis



- If the survivability analysis proves the design implementation does not satisfy mission requirements, user-inserted mitigation might be necessary.
 - This will change the design and its essential-bit count.
 - Essential-bit upper-bounds cannot be used to measure the survivability of applications with embedded mitigation.
 - Mitigation requires additional logic
 - Additional logic will increase the essential-bit count and consequently increase the estimated σ_{SEF} .
 - RTD-MFTF testing is required to measure the efficacy of the inserted mitigation. Can't assume mitigation performs as expected.
 - Requires the development team to perform SEU testing.
- Should analyze the design with-mitigation and without-mitigation (when possible)... used as another metric for the fidelity of the inserted mitigation.



Summary



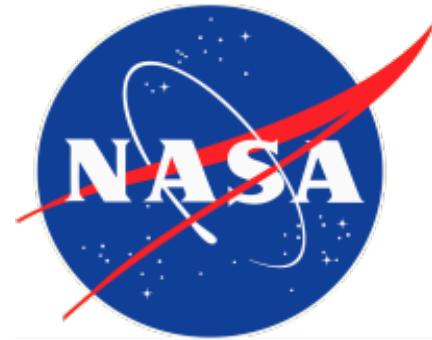
Single event transient(SET)

Single event latchup(SEL)

Single event functional interrupt (SEFI)

- Purpose of the work is to improve SEU data-sets used for survivability analysis.
- Generic SEU data obtained from testing simple structures (e.g., shift registers) are no longer adequate for SEU characterization of FPGA designs.
- An approach is presented that combines investigating simple and complex test structures:
 - Investigates the efficacy of using configuration SEU data with design specific information for survivability analysis.
 - Goal is to reduce the necessity of performing SEU testing on every design.
 - MFTF testing of complex structures is required to validate the approach (per SRAM-based FPGA family of devices).
- Xilinx Kintex UltraScale data are presented:
 - Data suggest that essential-bit SEU cross-section might be a reliable data-set for survivability analysis.
 - Additional testing by Xilinx is required and will be performed... yet initial results are promising.
 - Eventually, this approach can reduce the need for testing by the end-user.
- If mitigation is required, RTD-MFTF testing is required to be performed/orchestrated by the end-user.

Acknowledgements



Thank you for your support